

# HI-TIDE

## Release Notes for Version 3.15

Copyright (C) 2008 HI-TECH Software.  
All Rights Reserved. Printed in Australia.  
Produced on: June 13, 2008

HI-TECH Software Pty. Ltd.  
ACN 002 724 549  
45 Colebard Street West  
Acacia Ridge QLD 4110  
Australia

email: [hitech@htsoft.com](mailto:hitech@htsoft.com)  
web: <http://www.htsoft.com>  
ftp: <ftp://www.htsoft.com>

THIS FILE CONTAINS IMPORTANT INFORMATION RELATING TO  
HI-TIDE. PLEASE READ THIS DOCUMENT BEFORE RUNNING THE  
SOFTWARE.

# Chapter 1

## Description

HI-TIDE<sup>TM</sup> 3 (HI-TECH Software's Integrated Development Environment v3.15), based on the Eclipse Platform and CDT, provides an easy-to-use interface to HI-TECH C compilers and enhances code development.

The Eclipse Platform is an open-source IDE that provides an open platform for the integration of tools. CDT is an extension of the platform that provides generic support for C language development. Basing HI-TIDE<sup>TM</sup> on this platform allows HI-TECH Software to focus on developing unique embedded development tools, such as hardware debuggers, simulators, and code-generation tools for microcontroller peripherals. The Eclipse platform brings with it well-tested and widely used tools such as CVS integration, editor code completion, source outlines, etc.

## Chapter 2

# New Features

### 2.1 Eclipse & CDT Upgrade

The previous version of HI-TIDE (3.13) was built upon Eclipse 3.12 and CDT 2. This Beta Release of HI-TIDE has been built upon the Europa Fall release of Eclipse and CDT 4.0.3.

### 2.2 Changed New Project Wizard

This release of HI-TIDE provides easy access to a generic **HI-TECH C Project** wizard which is then used to create a project for any of the products you may have installed.

### 2.3 Support for JAVA SE 6 (1.6)

Some testing has been performed using Sun's JAVA SE 6 and has proved usable. Whilst JAVA SE 5 is recommended where possible, HI-TIDE will run on the newer runtime environments.

### 2.4 HI-TECH C JTAG Debug Interface Support

#### 2.4.1 Debugging

This release of HI-TIDE supports the HI-TECH C JTAG Debug Interface for the PIC32 revision C silicon.

To use the JTAG Debug Interface perform the following steps. First create a project, then select **Run->Debug...** from the menu bar. In the **Debug** dialog that appears select the launch for the project. Project launches are listed on the left hand side of the dialog. Typically the launch will be called *project\_name* **Launch** where *project\_name* is the name of the project. Once the launch is selected, select the **Debugger** tab, then select

the **JTAG Debugger** from the **Debugger:** combo box. Finally run the debugger by pressing the **Debug** button at the bottom right corner of the dialog.

For information on how to use the debugger follow the **Using the debugger and simulator** cheat sheet. This cheat sheet can be found by selecting **Help->Cheat Sheets...** from the menu bar. Then, in the dialog that appears, expand the **HI-TIDE Cheatsheets** folder and select **Debugger and Simulator**, after this select **OK**.

## 2.4.2 Programming

The HI-TECH C JTAG debugger interface can also be used via HI-TIDE to program and erase PIC32 devices.

To erase a PIC32 device select **Tools->MTAP Chip Erase** from the menu. This will execute a Microchip TAP (MTAP) ERASE CHIP command.

To program a PIC32 device with the contents of any Intel HEX file select from the menu **Tools->Flash Hex File** and the **Flash Utility** dialog will appear. Enter in the text field provided a path to the desired HEX file or click **Browse** to search for the file. Once a HEX file has been supplied select the target chip and click **Program** to begin programming the device.

## 2.5 PIC32 Simulator

This release of HI-TIDE contains a core simulator of the PIC32 architecture. The simulator features support for the following aspects of the PIC32 architecture:

- most of the MIPS32r2 and MIPS16e instructions including those generated by the HI-TECH C PRO for the PIC32 of the PIC32 compiler;
- all general purpose and system coprocessor registers;
- address error exceptions (load/store and instruction fetch);
- bus error exceptions (data reference: load or store);
- software trap exceptions;
- limited support for SFR clear/set/invert addresses (see section 3.4).

For information on how to use the simulator follow the **Using the debugger and simulator** cheat sheet. This cheat sheet can be found by selecting **Help->Cheat Sheets...** In the dialog that appears expand the **HI-TIDE Cheatsheets** folder and select **Debugger and Simulator** and then **OK**.

## 2.6 C-Wiz

This release of HI-TIDE contains a C-Wiz support PIC32 architecture and peripherals. This support includes the configuration/initialisation of the:

- Configuration registers and ID locations;
- 10 bit 16 Channel Analog to Digital Converter;
- Analog Comparator Module;
- Comparator Voltage Reference;
- Input Capture Module;
- Output Compare Module;
- I/O Ports;
- Timers 1 (type A) and 2-5 (type B).

## Chapter 3

# Limitations

### 3.1 Installation

This release of HI-TIDE is a stand-alone product from previous HI-TIDE releases. It will not inherit any configuration settings, user preferences or workspaces. Furthermore this release is incompatible with any previous workspaces and may cause corruption if switched to.

### 3.2 Debugger UI

**Variable View** The address contained in function pointers is displayed correctly, but the symbolic function name associated with this address is not displayed.

### 3.3 HI-TECH C JTAG Debug Interface Debugger for the PIC32

**Requires Power Cycle Before Launch** Before each launch of the JTAG debugger you must cycle the power of the attached development board.

**Slow C Stepping** C stepping can be slow. Especially when stepping over function calls.

### 3.4 PIC32 Simulator

**WRPGPR Not Simulated** The WRPGPR instruction has been implemented as a NOP. The instruction will also appear as a NOP in the disassembly view.

**SFR Clear/Set/Invert Addresses** Operations with these addresses work but in some cases they are not displayed correctly. When writing to an SFR the value is displayed in the memory view (under the DATA tab) at the SFR address correctly.

However, when viewing the SFR in the registers view only the last 8 bits are shown.

**Error Exceptions Not Produced** The following types of exceptions are not simulated:

- Bus error (instruction fetch)
- Syscall
- Breakpoint
- Reserved Instruction
- Coprocessor Unusable
- Arithmetic Overflow

**Interrupts Not Simulated** Interrupts in general are not simulated.

### 3.5 C-Wiz

**Limited PIC32 Support** The following peripherals are not supported by C-Wiz in this release:

- Inter-Integrated Circuit (I<sup>2</sup>C) Module;
- Parallel Master Port (PMP);
- Serial Peripheral Interface (SPI) Module;
- Universal Asynchronous Receiver Transmitter (UART) module.

## Chapter 4

# Licensing

- Visit <http://www.eclipse.org> for licensing information for the Eclipse Public License.
- The *Sun* Java Runtime Environment<sup>TM</sup> - License can be found in the file `sun-jre-license.txt` in the HI-TIDE<sup>TM</sup> 3 distribution's docs directory.
- DOM4J<sup>TM</sup> - License can be found in the file `dom4j-license.txt` in the HI-TIDE<sup>TM</sup> 3 distribution's docs directory.
- This product also uses source code provided by *Sun*. This source code is covered under the license found in the file `sun-samples-license.txt`.